УДК 004.934

В.Г. Прохоров

ИСПОЛЬЗОВАНИЕ КАРТ КОХОНЕНА ДЛЯ УСКОРЕНИЯ ФРАКТАЛЬНОГО СЖАТИЯ ИЗОБРАЖЕНИЙ

Рассмотрены основы классического алгоритма фрактального сжатия изображений, описаны его недостатки. Предложены алгоритмы, ускоряющие фрактальное сжатие за счет выделения характеристик, и с помощью карт Кохонена. Приведены экспериментальные данные, иллюстрирующие эффективность предложенного алгоритма.

Введение

Алгоритм фрактального сжатия изображений является одной из наиболее перспективных альтернатив алгоритмам, использующих дискретное косинусоидальное преобразование (ДКП), например, JPEG. В силу своих преимуществ, именно фрактальное сжатие применяется в ряде узкоспециализированных задач, важнейшей из которых является передача изображений со спутников (в этом случае, преимущество достигается за счет лучшего качества изображений, сжатых с коэффициентом компрессии больше 100, по сравнению с ДКП алгоритмами). Еще одно преимущество – является возможность применения к сжатым изображениям фрактального масштабирования и фрактальной интерполяции [1].

Широкое применение фрактального метода к сжатию изображений сдерживается его большой вычислительной сложностью, и, в некоторых случаях, ассиметричностью алгоритма. В отличии от алгоритмов ДКП, в которых происходит однозначное преобразование элементов изображения, во фрактальном алгоритме происходит поиск максимально соответствующих частей изображения за счет их полного перебора и попиксельного сравнения. Алгоритм фрактального кодирования, его преимущества и недостатки будут подробно описаны далее.

Проблеме ускорения работы фрактального сжатия и оптимизации перебора посвящено большинство статей, исследовавших этот алгоритм. Наиболее эффективными оказались два направления исследований: метод выделения особенностей (feature extraction) и метод классификации доменов (classification of domains). Первый метод заменяет попиксельное сравнение частей изображения сравнением векторов его характеристик (заранее вычисленных и нормированных). Второй метод, использующий карты Кохонена – его логическое продолжение, и значительно уменьшает число операций перебора доменных блоков.

Классический алгоритм фрактальной компрессии изображений

Основа метода фрактального кодирования — это обнаружение самоподобных участков в изображении. Впервые возможность применения теории систем итерируемых функций (IFS) к проблеме сжатия изображения была исследована Майклом Барнсли [2] (Michael Barnsley) и Аланом Слоуном (Alan Sloan), которые запатентовали свою идею в 1990 и 1991 гг.. Джеквин (Jacquin) представил метод фрактального кодирования, в котором используются системы доменных и ранговых блоков изображения (domain and range subimage blocks), блоков квадратной формы, покрывающих все изображение. Этот подход стал основой для большинства методов фрактального кодирования, применяемых сегодня. Он был усовершенствован Ювалом Фишером (Yuval Fisher) и другими исследователями.

В соответствии с данным методом изображение разбивается на множество неперекрывающихся ранговых подизображений (range subimages) и определяется множество перекрывающихся доменных подизображений (domain subimages). Для каждого рангового блока алгоритм коди-

рования находит наиболее подходящий доменный блок и аффинное преобразование, которое переводит этот доменный блок в данный ранговый блок. Структура изображения отображается в систему ранговых блоков, доменных блоков и преобразований.

Идея фрактального сжатия использует представление исходного изображения в качестве неподвижной точкой некоего сжимающего отображения. Тогда можно вместо самого изображения запомнить каким-либо образом это отображение, а для восстановления достаточно многократно применить это отображение к любому стартовому изображению, поскольку по теореме Банаха, такие итерации всегда приводят к неподвижной точке, т. е. к исходному изображению. На практике вся трудность заключается в отыскании по изображению наиболее подходящего сжимающего отображения и в компактном его хранении. Как правило, алгоритмы поиска отображения (т. е. алгоритмы сжатия) в значительной степени переборные и требуют больших вычислительных затрат. При этом, алгоритмы восстановления достаточно эффективны и быстры.

Вкратце метод, предложенный Барнсли, можно описать следующим образом. Изображение кодируется несколькими простыми преобразованиями (в нашем случае аффинными), т. е. определяется коэффициентами этих преобразований (в нашем случае, каждое преобразование описывается шестью коэффициентами, из них 4 коэффициента определяют угол поворота и фактор масштабирования, а 2 – величину пространственного переноса). Например, изображение кривой Коха можно закодировать четырьмя аффинными преобразованиями, т. е. эту фигуру можно однозначно определить с помощью всего 24-х коэффициентов.

Для декомпрессии изображения можно, поставив чёрную точку в любой точке картинки, применять преобразования в случайном порядке некоторое (достаточно большое) число раз (этот метод ещё называют фрактальный пинг-понг). В результате, будет происходить поблочное восстановление картинки.

Основная сложность метода

Основная сложность фрактального сжатия заключается в том, что для нахождения соответствующих доменных блоков требуется полный перебор. Поскольку при этом переборе каждый раз должны сравниваться два массива пикселов, данная операция является медленной. Для ее ускорения в экспериментальной части данной работы применялся ряд ограничений, что позволило несколько сократить объем вычислений.

1. Все области являются квадратами со сторонами, параллельными сторонам изображения. Это ограничение достаточно жесткое. Фактически все многообразие геометрических фигур аппроксимируется лишь квадратами.
2. Размер доменного блока равен 8 пикселам. При переводе доменной области в ранговую уменьшение размеров производится ровно в два раза, т. е. размер рангового блока равен 4 пикселам. Это существенно упрощает как компрессор, так и декомпрессор, так как. задача масштабирования небольших областей является несложной.
3. Все доменные блоки – квадраты и имеют фиксированный размер. Изображение равномерной сеткой разбивается на набор доменных блоков. (Метод квадродерева не применяется).
4. Доменные области брались с половинным перекрытием по вертикали и по горизонтали.
5. При переводе доменной области в ранговую поворот куба возможен только на 90°, 180° или 270° градусов. Также допускается зеркальное отражение. Общее число возможных преобразований (считая пустое) – 8

Для того, чтоб лучше понять проблему вычислительной сложности фрактальной компрессии, рассмотрим особенности фрактального компрессора, используемого в экспериментальной части данной работы. Сжатию подвергались цветные изображения размером 256 на 256 пикселов в формате BMP. Кроме коэффициентов аффинного преобразования, в структуре сжатого файла сохранялось также оптимальное изменение цвета (оп-

тимальный сдвиг по яркости). Такой метод применяется в случае, когда сжатию подлежат реальные изображения, подбор оптимальной яркости доменного блока для наилучшего соответствия ранговому блоку позволяет одновременно уменьшить число вычислений, и, зачастую, найти более подходящее соответствие. Это можно проиллюстрировать на примере. Допустим, происходит поиск наилучшего домена для монолитно серого рангового блока. В этом случае, ему может соответствовать любой другой монолитный домен, если при декомпрессии учесть изменение яркости.

Сам алгоритм упаковки сводится к перебору всех ранговых блоков и подбору для каждого соответствующего ему доменного. Для каждого рангового блока делаем его проверку со всеми возможными доменными блоками (в том числе с прошедшими преобразование симметрии), находим вариант с наименьшей мерой, в данном случае – среднеквадратичным отклонением, и сохраняем коэффициенты этого преобразования в файл. Коэффициенты – это (1) координаты найденного блока, (2) число от 0 до 7, характеризующее преобразование симметрии (поворот, отражение блока), и (3) сдвиг по яркости для этой пары блоков. Сдвиг по яркости вычисляется как:

$$q = \bar{r} - \left(\frac{\alpha}{\beta}\right)\bar{d}, \qquad (1)$$

где $r_{i,j}$ – значения пикселов рангового блока; $d_{i,j}$ – значения пикселов доменного блока, $\bar{r}, \bar{d}$ – средние значения пикселов ранговых и доменных блоков; α – величина корреляции рангового и доменного блоков; β – дисперсия значений пикселов доменного блока. При этом мера считается как:

$$d(R,D) = \sum_{i=1}^{n}\sum_{j=1}^{n}(0.75r_{ij} + q - d_{ij})^2, \qquad (2)$$

Мы не вычисляем квадратного корня из меры и не делим ее на n, поскольку данные преобразования монотонны и не помешают нам найти экстремум, однако мы сможем выполнять на две операции меньше для каждого блока.

Очевидно, что наибольшее число операций происходит при расчетах формул (1) и (2). В случае полного перебора необходимо выполнить $N_D * N_R$ вычислений по формулам (1) и (2), где N_D, N_R – количество доменных и ранговых блоков соответственно. Заметим, что каждый из таких расчетов – ресурсоемкий блок вычислений.

Уменьшение общего числа вычислений может быть достигнуто за счет уменьшения общего числа сравнения блоков, за счет замен формул (1) и (2) более эффективными, с вычислительной точки зрения, формулами, либо комбинацией таких подходов.

Использование метода вычисления характеристик для ускорения сжатия

Основная идея метода – выделение для каждого доменного и рангового блока небольшого набора характеристик, описывающих этот блок. Такие характеристики вычисляются один раз при формировании массивов доменных и ранговых блоков. С учетом полученных характеристик, мерой при сравнении блоков будет являться евклидово расстояние между векторами признаков

$$d(R,D) = \sqrt{\sum_{i=1}^{n}(r_i - d_i)^2}, \qquad (3)$$

где r_i, d_i – соответствующие характеристики рангового и доменного блока, n – число характеристик каждого блока. Принципиальное отличие классического алгоритма фрактальной компрессии, и алгоритма, использующего характеристики состоит в том, что последний вычисляет сдвиг яркости всего один раз, когда наиболее подходящий домен уже найден. Классический алгоритм вычисляет сдвиг по яркости при каждом сравнении блоков. Таким образом, алгоритм, использующий выделение характеристик, не проводит вычислений по формуле (1), а за счет этого получает выигрыш в производительности.

Вопрос выбора набора характеристик является открытым. Такими характеристиками могут быть результаты спектрального анализа Фурье [3], вейвлет анализа [4], характеристики оттенка или текстуры изображения [5]. Желательно, чтоб выбранные характеристики были инвариантны к повороту изображения, и не повторялись при применении аффинных преобразований. При написании данной работы блок представлялся пятью характеристиками: стандартное отклонение, асимметрия, межпиксельная контрастность, бета, максимальный градиент, который является максимумом из вертикального и горизонтального градиентов изображения. Поскольку числовые значения этих характеристик значительно отличаются, необходимо провести предварительное нормирование таких значений.

Ускорения сжатия с помощью самоорганизовующейся карты Кохонена

Большое время кодирования– это результат того, что приходится производить большое количество доменно-ранговых сопоставлений, а это наиболее затратная часть работы компрессора.

Вышеописанный алгоритм с выделением характеристик сокращает время кодирования путем замены попиксельной обработки более простым сопоставлением характеристик. Только те домены, которые прошли через сопоставление характеристик, задействуются в попиксельном сопоставлении. Таким образом, большой объем вычислений, связанных с доменно-ранговым сопоставлением, исключается за счет предварительного сопоставления характеристик.

Следующий шаг, который можно использовать для уменьшения объема вычислений, связанных с доменно-ранговыми сопоставлениями – это классификация доменных и ранговых областей. Тогда доменно-ранговые сопоставления выполняются только для тех доменов, которые принадлежат классу подобия данной ранговой области. По сути, вышеописанный метод выделения особенностей является разновидностью схемы классификации. Вычисление характеристик служит для определения тех доменов, которые принадлежат классу изображений, чьи вектора характеристик не выходят за пределы допуска для вектора характеристик данного рангового блока. Более сложные схемы классификации используют заранее определенное множество классов. Алгоритм классификации связывает каждый домен с одним из этих классов. При кодировании алгоритм связывает данный ранговый блок с определенным классом, и после этого доменно-ранговое сопоставление проводится только с доменами, отнесенными к этому классу (и, возможно, с другими подобными классами). Сбережение времени при кодировании происходит за счет выполнения меньшего числа доменно-ранговых сопоставлений. Основой описанной здесь схемы классификации является самоорганизующаяся нейронная сеть, которая обучается на данных векторов характеристик, полученных для доменных блоков некоторого стандартного изображения. Это те пять характеристик, которые были описаны в алгоритме, использующем характеристики блоков. Преимущество использования самоорганизующейся сети в том, что не нужно решать, какой класс должен быть выбран при классификации. Сеть самоорганизуется в кластеры, представляющие классы изображений, которые определяются содержащимися в них данными изображения. Более того, изображение, используемое для обучения, может не быть (и, как правило, не является) подобным тому изображению, которое должно быть закодировано. Таким образом, время обучения сети не входит в общее время кодирования. При небольшом количестве доменов классификационный подход не дает существенного выигрыша во времени по сравнению с характеристическим подходом. Однако для большого количества доменов (20000 и более) выигрыш во времени значителен, а большое количество доменов может обеспечить высокое качество изображения при данном коэффициенте сжатия [6].

Самоорганизующиеся нейронные сети (называемые по имени изобретателя картами Кохонена) характеризуются многомерным массивом, или решеткой узлов. С каждым узлом решетки связан весовой вектор, который имеет ту же размерность, что и входные векторы, которые будут использованы для обучения. Размерность решетки не обязательно должна быть такой как размерность весового вектора. Сложные взаимосвязи во многих задачах требуют, чтобы размерность весовых векторов была высокой, поэтому для корректной самоорганизации нужна решетка с высокой размерностью.

Процесс обучения сети происходит абсолютно независимо, без какого-либо контроля со стороны человека. Сначала сеть весовых векторов инициализируется случайными значениями. Затем в сеть поступает входной вектор характеристик, и ищется весовой вектор, самый близкий к входному вектору. Затем находится узел, весовые коэффициенты которого находятся ближе всего ко входному вектору (в качестве метрики используется евклидово расстояние). Веса, соседние по решетке с выбранным узлом (этот узел называют «узел-победитель») адаптируются, чтоб больше походить на входной вектор.

$$w_{ij}^{новый} = w_{ij}^{старый} + \varepsilon(v - w_{ij}^{старый}), \qquad (4)$$

где i, j – индексы узлов, смежных с победителем. В некоторых вариациях самоорганизовующихся сетей, размер окрестности, в которой ищутся такие узлы, сокращается с каждой новой итерацией процесса обучения. Параметр ε – скорость обучения, такой параметр используется при обучении большинства нейросетей. В качестве формулы (4) можно также использовать видоизмененную функцию Гаусса. Такой подход лучше подходит для сложных случаев, когда по тем или иным причинам, сеть не самоорганизовуется. При выполнении данной работы такового не наблюдалось, поэтому использовался более быстрый вариант обучения.

Необходимо отметить, что решетка помогает определить топологию окрестности для весовых векторов. Весовые векторы, относящиеся к узлам решетки, близко расположенным друг к другу, будут иметь свойства, которые являются подобными относительно распределения характеристик в исходных данных. Топология окрестности используется в процессе поиска домена.

Рассмотрим особенности карты Кохонена, используемые для ускорения фрактального сжатия. Размерность векторов характеристик, используемых здесь, равна 5, а решетка состоит из 8 х 8 узлов. Каждый узел решетки представляет класс доменных блоков, поэтому выбор количества узлов является очень важным. С одной стороны, чем больше узлов, тем больший потенциальный выигрыш можно получить в скорости работы. Но значительное число узлов ухудшает способность сети обобщать входные данные, а также тормозит обучение. Очевидно, что наилучший вариант – нахождение компромисса между этими двумя противоречиями. В данной работе сеть содержит 64 узла, так как при этом соблюдается квадратная топология решетки, и при этом сохраняется баланс между скоростью обучения, способностью обобщать и приростом производительности. Начальный размер окрестности смежных узлов должен быть приблизительно равен половине размера строк и столбцов, в данном случае 4. Если размер окрестности будет слишком мал, то это может привести к плохой топологии окрестности (т. е. несхожие весовые векторы могут оказаться в решетке близко друг к другу).

После завершения обучения, доменные блоки для данного изображения классифицируются путем сопоставления каждому из них весового вектора, самого близкого к нему в пространстве характеристик. Таким образом, каждый узел решетки теперь имеет весовой вектор и связанный с ним список доменных блоков. Этот список доменных блоков принадлежит классу, сопоставленному этому весовому вектору. Класс – это, по определению, множество всех изображений, более близких в пространстве характеристик к

данному весовому вектору, чем к какому-либо другому вектору в решетке.

Когда вектор характеристик рангового блока поступает в сеть, ему точно так же сопоставляется весовой вектор сети. Затем ранговый блок сравнивается с доменами, которые сопоставлены данному весовому вектору, а также с доменами, сопоставленными весовым векторам из окрестности данного весового вектора в сети. Параметр, определяющий максимальный радиус поиска, задает размер окрестности. Как и раньше, происходит поиск наилучшей (т. е. минимальной) на данный момент разности между характеристиками. Если новый домен в списке обеспечивает меньшую разность, происходит полное попиксельное доменно-ранговое сопоставление. Единственная разница между данным алгоритмом и характеристическим алгоритмом, в том, что количество доменов, для которых выполняется вычисление характеристического расстояния, в данном алгоритме меньше. Данный алгоритм имеет некоторые вычислительные издержки, связанные с поиском весового вектора, ближайшего к характеристическому вектору рангового блока (это серия вычислений характеристического расстояния). Значение этих издержек уменьшается с увеличением общего количества доменов.

Экспериментальная оценка эффективности алгоритмов

Для оценки эффективности предложенного подхода была разработана программа, реализующая фрактальное сжатие каждым из трех предложенных алгоритмов.

Все три алгоритма были настроены на поиск наилучшего соответствия доменных и ранговых блоков. Это несколько отличается от реальных реализаций фрактальных компрессоров, где поиск доменно-рангового соответствия прекращается после нахождения первого подходящего домена (в этом случае, при сравнении используется определенное пороговое значение ошибки). На вход поступало цветное изображение размером 256 x 256 пикселов, домены брались из шаблонного изображения размером 500 x 375 пикселов. Общее число ранговых блоков равнялось 4096, доменных – 90528.

Результаты экспериментов приведены в таблице. Стоит отметить, что в общее время сжатия с помощью карт Кохонена не включалось время обучения сети, равное 10:49 минутам. Такое действие делается один раз, поэтому включить его во время сжатия было бы некорректно. Аналогично, время работы характеристического алгоритма включает в себя расчет характеристик и нормирование только для ранговых блоков – для доменов эти действия выполняются один раз, после чего результаты сохраняются.

Что касается непосредственных результатов эксперимента, то они подтверждают эффективность использования карт Кохонена в оптимизации тех вычислений, где присутствует перебор различных данных и поиск наилучшего соответствия.

Таблица. Результаты экспериментов

Тип алгоритма	Время сжатия, мин/сек.
Классический алгоритм	24:35
Характеристический алгоритм	4:06
Карта Кохонена	0:31

Выводы

Задача ускорения фрактального сжатия все еще остается актуальной. В своих новейших работах Хайкин [7] и Фишер [8] рассматривают другие методы кластеризации, в которых основное внимание уделяется возможностям проведения быстрого дообучения с учетом новых примеров доменных блоков.

Тем не менее, основной задачей данной работы было показать то, как средствами исскуственного интеллекта (в данном случае, ими являются самоорганизовующиеся карты Кохонена) достигается ускорение вычислений. Проведя кластеризацию, простой перебор всех данных заменяется сравнением с отдельным подклассом этих данных. Еще одним важным преимуществом является обучаемость карт Кохонена без учителя: в отличии от классических полносвязных нейросетей, на вход которых поступает вектор и целевое значение, сети Кохонена способны разбивать информацию на классы даже если такого целевого значения нет, что часто встречается в реальных задачах. С другой стороны, карты Кохонена никогда не смогут заменить классические и сверточные нейросети в задачах распознавания, из-за своей плохой способности точно разделять линейно неразделимые данные. Именно поэтому, в некоторых реальных задачах перебор осуществляется не только с классом узла победителя, но и близлежащими узлами.

1. *Chuan-jiang He, Xiao-na Shen, Gao-ping Li.* Interpolation decoding method with variable parameters for fractal image compression, Chongquing University . – 2005. – P. 1 – 5.
2. *Barnsley M.* Fractals everywhere – Boston, Academic Press, 1993. – P. 103 – 114.
3. *McGregor D.* Fast fractal transform method forDataCompression, University of Strathclyde. – 1994. – P. 1 – 8.
4. *Herbert D.* Fast Fractal Image Compression with Triangulation Wavelets, IEEE Press . – 1998.
5. *Welstead, S.* Self-Organizing Neural Network Domain Classification for Fractal Image Coding, Speech and Image Processing Services Research AT&T Lab. – 1997. – P. 248 – 251.
6. *Hamzaoui R.* Codebook clustering for Self-Organizing Maps for Fractal Image Compression, NATO Advanced Study Institute. – 1995.
7. *Хайкин С.* Нейронные сети. Полный курс Изд. второе (исправленное). Прэнтис Холл. – 2006. – С. 239 – 298 ; 308 – 315.
8. *Fisher Y.* Fractal Image Compression. – Springer-Verlag. – 1998.

Получено 06.03.2009

Об авторе:

Прохоров Валерий Георгиевич,
аспирант Института программных систем НАН Украины.

Место работы автора:

Институт программных систем
НАН Украины.
03187, Киев -187, проспект Академика Глушкова, 40.
Телефон: 80509713876.
E-mail: makumazan84@yahoo.com